HARVEY®

HARVEY
H-Text Manual

Version 2.7.0

DSPECIALISTS
Digitale Audio- und Messsysteme GmbH
Helmholtzstr. 2-9 L, D-10587 Berlin

http://www.dspecialists.com
http://www.harvey-audio.com

History

Version	Date of Change	Description
1.0	01.10.2014	Release of H-Text for V1.4 Composer & V1.5 Firmware
1.9.1	27.04.2017	Added SUB/NFY mechanism as released in HARVEY firmware 1.9.1.
2.1.4	25.10.2018	Updated text for new HARVEY device generation.
2.3.0	23.09.2019	Added Input Selector Block
2.5.0	02.04.2020	Added STORE command for presets
2.6.0	04.06.2020	Added BEGIN/END commands for H-Text transactions
2.7.0	06.10.2020	Marked older H-Net protocol as deprecated in Introduction

Contents

1 Introduction

HARVEY can be controlled via the control dialogs in the HARVEY Composer software and via third-party controllers using any of HARVEY´s serial or network control interfaces (RS-232, RS-485 or TCP/IP Ethernet).

HARVEY integrates two different types of control protocols:

1. H-Text: Human readable, string based, available since firmware V1.5 / GUI V1.4.
2. H-Net: More technical, byte-oriented. Deprecated.[1]

Both protocols enable users to integrate the controllable features of HARVEY into remote controls (e.g. by AMX, Crestron or other touch panel manufacturers). Controllable features are for instance parameters of functional processing blocks (e.g. gains, EQ settings), preset calls and the retrieval of current signal levels.

This document brings into focus the H-Text functionality of HARVEY and intends to enable system integrators to use it in media control applications.

2 Compatibility

To keep things simple, the HARVEY products and manuals use a consistent versioning scheme that indicates compatibility.

The version of this document can be found on the title page. It tells, which version of HARVEY firmware is at least required to support *all* H-Text features described in the following chapters.

Note, that the H-Text protocol is designed to be *backward compatible.* New HARVEY firmware versions should always be fully compatible to older specifications of the H-Text protocol.

Should there ever be a need for breaking changes to the protocol, those will be prominently communicated.

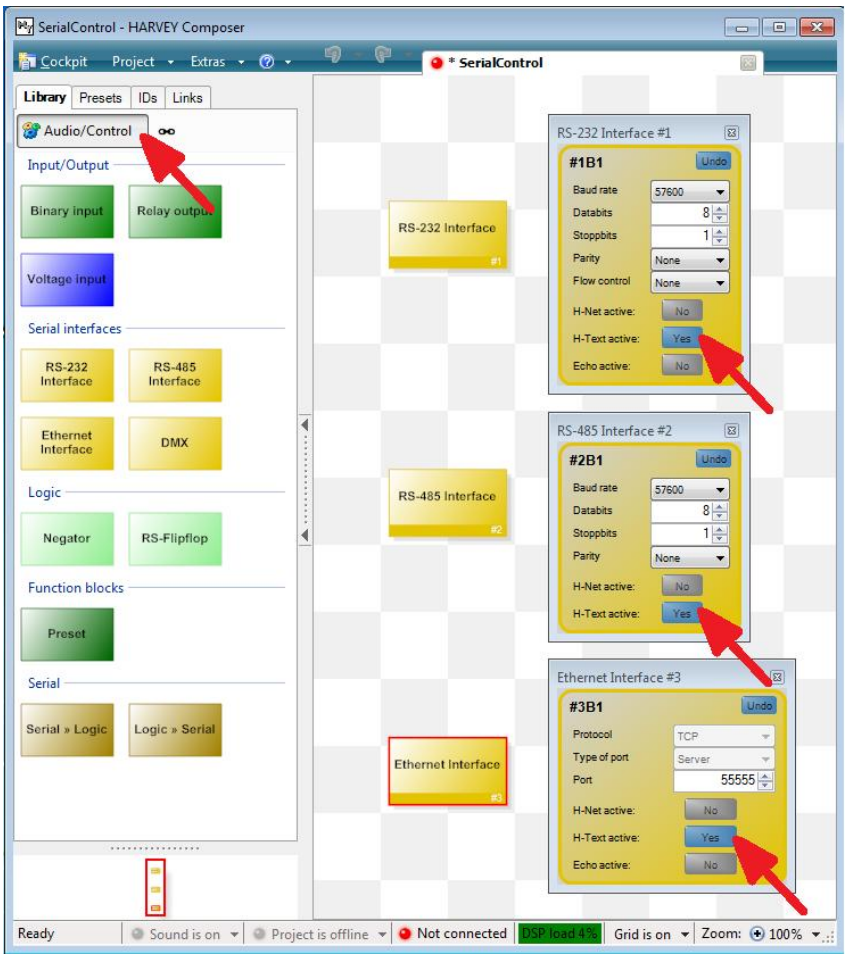
[1] Note: The older H-Net protocol is *not* recommended for use in new control applications. It is only kept in the firmware for compatibility reasons and does not receive any new features or improvements.

3 Usage in HARVEY Composer

HARVEY offers the control functionality via its RS-485, RS-232 and 10/100BaseT Ethernet interface.

Though the protocol payload is identical and fixed for all interface types, the user has to parameterize UART-parameters for RS-232 and RS-485 and network protocol parameters for Ethernet in HARVEY Composer. The interface's parameters have to be defined in HARVEY Composer on the control layer (switch by toggling “Audio/Control”):

Additionally the H-Text protocol parser has to be enabled in the appropriate serial interface block within your HARVEY Composer project.



3.1 RS-232 and RS-485 Interface parameters

Baud rate, data bits, stop bits and parity chosen in HARVEY Composer have to be used identically by the remote control. Default parameters are 57600 bits per second, 1 start bit, 8 data bits, 1 stop bit, no parity.

Notice: The “H-Text active” button needs to be switched to “Yes” in order to activate the H-Text parser on this interface.

Notice: If used in combination with a RS-232/RS-485 converter please disable flow control!

3.2 Ethernet Interface parameters

Currently the transport protocol is limited to TCP and server functionality on any valid port number. Default parameters are TCP, server and port 55555.

Notice: The “H-Text active” button needs to be switched to “Yes” in order to activate the H-Text parser on this interface.

Notice: There can be multiple Ethernet interfaces with an active H-Net parser, as long as they use different port numbers.

4 H-Text Protocol Reference

4.1 H-Text: Overview

H-Text follows a simple request-response handshake: A controller sends a command request to a HARVEY device which executes the command and answers with a response message (see section 4.9).

A H-Text command string to be sent by a controller has this basic structure:

```
Command [DeviceNumber] Attribute InstanceID Index1 Index2 Value <LF>
```

A correct command string is answered by a response message from HARVEY:

```
Response <CR> <LF>
```

- “Command” defines the action to be performed on Attribute → Section 4.2
- “DeviceNumber” is used to identify the HARVEY device. The “DeviceNumber” parameter is optional. → Section 4.3
- “Attribute” identifies the parameter type to be addressed. → Section 4.5
- “InstanceID” identifies the specific functional block or preset in a HARVEY Composer project to be addressed. → Section 4.6
- “Index1” and “Index2” are used for some attributes to address a specific element within a functional block (e.g. a matrix mixer node or equalizer band). → Section 4.7
- In a SET command string “Value” represents the content to be applied to the attribute. The representation of numeric values depends on the attribute.

  In an INC or DEC command string “Value” defines the step width the attribute is to be incremented or decremented. → Section 4.8
- The mandatory line feed <LF> (ASCII: 10dec) character terminates the command string and optionally may be preceded by a carriage return <CR> (ASCII: 13dec) character.
- Correct command strings are answered by a “Response”. → Section 4.9

4.1.1 H-Text: Example Command Messages

In following examples “*blank*” represents an empty space within the command structure.

Example: String to set a Level Gain to -20 dB:

```
SET 1 LVLGAIN #2B5 -200 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	1	LVLGAIN	#2B5	*blank*	*blank*	-200	<LF>

Example: String to mute a Matrix Mixer Crosspoint on any device (device number omitted):

```
SET MXXPMUTE #10B1 3 5 1 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	*blank*	MXXPMUTE	#10B1	3	5	1	<LF>

Example: String to call a Preset 5:

```
RECALL 3 PRESET 5 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
RECALL	3	PRESET	5	*blank*	*blank*	*blank*	<LF>

Example: String to update the parameter values that are stored in Preset 5:

```
STORE 3 PRESET 5 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
STORE	3	PRESET	5	*blank*	*blank*	*blank*	<LF>

Example: Subscribe to Level Gain value of Block #2B5:

```
SUB 1 LVLGAIN #2B5 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SUB	1	LVLGAIN	#2B5	*blank*	*blank*	*blank*	<LF>

4.2 H-Text: Commands

Following H-Text commands are available:

`SET`	Apply a new value to a parameter.
`GET`	Retrieve a parameter value.
`SUB`	Subscribe to changes of a parameter value
`UNSUB`	Unsubscribe previously subscribed value changes
`INC`	Increment parameter value by a step width.
`DEC`	Decrement parameter value by a step width.
`RECALL`	Recall stored preset.
`STORE`	Store current parameter settings to preset
`BEGIN`	Start H-Text transaction
`END`	Commit H-Text transaction

Notice: Values that exceed the defined range of a parameter in a SET command may be cropped automatically by the HARVEY device to the respective minimum or maximum value. That will also be done in INC or DEC commands if the resulting parameter value would exceed the allowed range.

4.3 H-Text: Device Number

DeviceNumber is used to identify the HARVEY device in the H-Text and H-Net protocols.

The DeviceNumber is *optional* and may be left out to address *any* device. That may especially be useful on point-to-point connections (RS232, TCP/IP) where the device identification is implied by the connection.

The default DeviceNumber is 1.

The DeviceNumber may be accessed using the SET or GET command via the DEVNUM system attribute. See section 0.

Example: H-Text string to get the device number of all listening device

```
GET DEVNUM <LF>
```

Example: H-Text string for changing the device number of device 3 to 11

```
SET 3 DEVNUM 11 <LF>
```

4.4 H-Text: Transactions (Multiple command execution)

H-Text Transactions provide clients with a mechanism to significantly speed up execution of multiple commands.

Example use cases:

- Subscribing multiple parameters after connect.
- Setting multiple matrix nodes.

The H-Text keywords BEGIN and END are used to encapsulate the set of commands to be executed.

Example: Sequence of commands to set 6 matrix nodes

```
BEGIN SET <LF>

SET MXXPMUTE #1B1 1 1 0 <LF>
SET MXXPMUTE #1B1 1 2 1 <LF>
SET MXXPMUTE #1B1 1 3 1 <LF>
SET MXXPMUTE #1B1 2 1 1 <LF>
SET MXXPMUTE #1B1 2 2 0 <LF>
SET MXXPMUTE #1B1 2 3 1 <LF>

END <LF>
```

Example: Sequence of commands to subscribe multiple project parameters

```
BEGIN SUB <LF>

SUB MXXPMUTE #1B1 1 1 <LF>
SUB MXXPMUTE #1B1 1 2 <LF>
SUB METERLVL #7B1 <LF>
SUB METERLVL #7B2 <LF>
SUB LVLGAIN #6B1 <LF>
SUB LVLGAIN #5B1 <LF>
SUB INPSEL #8B1 <LF>

END <LF>
```

The benefit of this syntax is that existing H-Text code can be easily extended to make use of transactions. Only the BEGIN and END statements have to be added to wrap existing SET or SUB commands.

H-Text transactions are dedicated to one type of commands, i.e. SET, SUB, or UNSUB. The intended type is expressed after the BEGIN keyword. All following commands that are not of that type will not be treated as part of the transaction.

All commands that have been recognized as part of the transaction will not be executed immediately and will not return an immediate result. Execution will be triggered upon receiving the END keyword.

Return values for BEGIN:

- OK, if resources for the transaction have been successfully allocated.
- ERR, if transaction type is missing or invalid
- BUSY, if resources could not be allocated (e.g. max number of transactions exceeded)

H-Text transactions return one aggregated result for all commands.

Return values for END:

- OK, if the transaction has been completely and successfully executed.
- ERR, if at least one command in the transaction failed
- BUSY, if HARVEY was not capable of processing commands at the time

Upon failure of a SET transaction, all parameters will be restored to their original value.

Limitations

- The number of transactions is limited to one per connection.
- Transactions are limited to 4000 commands each.
- Only commands SET, SUB, and UNSUB are supported in H-Text transactions.
- Failed transactions of type SUB may lead to some of the resources to remain subscribed after returning ERR.
- The type of commands (i.e. SET, SUB, etc.) may not be mixed within a transaction. E.g. if a SUB command is received within a SET transaction it will be executed immediately and not wait for the END command to be received.
- Transaction rollback may not always be possible or complete, depending on the cause of the failure or if logic events have fired that are directly hooked to a parameter change, which cannot not be undone.

4.5 H-Text: Attribute Reference per Functional Block

4.5.1 Analog Input Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Mic/Line	INPMICLINE	SET, GET, SUB	None	0 = Mic 1 = Line
Phantom Power	INPPHPWR	SET, GET, SUB	None	0 = off 1 = on
Mic Gain	INPMICGAIN	SET, GET, INC, DEC, SUB	None	0; 10..65
Line Gain	INPLINEGAIN	SET, GET, INC, DEC, SUB	None	0 = 0dB 1 = 6 dB 2 = 12 dB 3 = 18 dB

4.5.2 Analog Output Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Mute	OUTMUTE	SET, GET, SUB	None	0 = unmuted 1 = muted
Gain	OUTGAIN	SET, GET, SUB, INC, DEC	None	0 = -18 dB 1 = -15 dB 2 = -9 dB 3 = 0 dB

4.5.3 Level Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Gain	LVLGAIN	SET, GET, SUB, INC, DEC	None	-1000..+100 Stepwidth: 0.1 dB
Mute	LVLMUTE	SET, GET, SUB	None	0 = unmuted 1 = muted
PhaseReverse	LVLPHREV	SET, GET, SUB	None	0 = normal 1 = reversed

4.5.4 Level Meter Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Current Level	METERLVL	GET, SUB	None	-1000..+500 Stepwidth: 0.1 dB

4.5.5 AGC Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Bypass	AGCBYP	SET, GET, SUB	None	0 = AGC active 1 = AGC inactive

4.5.6 AVC Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Bypass	AVCBYP	SET, GET, SUB	None	0 = AVC active 1 = AVC inactive

4.5.7 Dynamic Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Bypass	DYNBYP	SET, GET, SUB	None	0 = Dynamics active 1 = Dynamics inactive

4.5.8 Ducker Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Bypass	DUCKBYP	SET, GET, SUB	None	0 = Ducker active 1 = Ducker inactive

4.5.9 Mixers

In Mixer commands, Index1 is used to address input channels and Index2 to address output channels. In a Matrix Mixer both indexes are used to address crosspoint parameters.

Example: String to mute the Matrix Mixer Crosspoint for input 3 and Output 5

```
SET MXXPMUTE #10B1 3 5 1 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	None	MXXPMUTE	#10B1	3	5	1	<LF>

Example: String to mute the Matrix Mixer Input 6 (for all outputs)

```
SET MXINMUTE #10B1 6 1 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	None	MXINMUTE	#10B1	6	0	1	<LF>

4.5.9.1 Matrix Mixer Block

Parameter of Functional Block	**Attribute String**	**Applicable Commands**	**Applicable Index**	**Applicable Value Range**
Input Gain	MXINGAIN	SET, GET, SUB, INC, DEC	Index1	-1000..+100 Stepwidth: 0.1 dB
Input Mute	MXINMUTE	SET, GET	Index1	0 = unmuted 1 = muted
Output Gain	MXOUTGAIN	SET, GET, SUB, INC, DEC	Index2	-1000..+100 Stepwidth: 0.1 dB
Output Mute	MXOUTMUTE	SET, GET, SUB	Index2	0 = unmuted 1 = muted
Crosspoint Delay	MXXPDELAY	SET, GET, SUB, INC, DEC	Index1, Index2	0..48000 [Samples]
Crosspoint Gain	MXXPGAIN	SET, GET, SUB, INC, DEC	Index1, Index2	-1000..+100 Stepwidth: 0.1 dB
Crosspoint Mute	MXXPMUTE	SET, GET, SUB	Index1, Index2	0 = unmuted 1 = muted

4.5.9.2 Auto Mixer Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Hold time	AMHOLDTIME	SET, GET, SUB	None	25..1000 stepwidth: 0.01 s
Max NoM	AMMAXNOM	SET, GET, SUB	None	0..Number of channels
Last Mic	AMLASTMIC	SET, GET, SUB	None	-1 = Last open Mic 0 = none 1..n = channel number
Input Threshold	AMINTHRES	SET, GET, SUB, INC, DEC	Index1	-1000..0 stepwidth: 0.1 dBFS
Input Priority	AMINPRIO	SET, GET, SUB	Index1	0 = priority off 1 = priority on

4.5.10 Input Selector Block

The input selector’s chosen input can be switched using this H-Text command.

Example: String to activate source 2 (1-based addressing)

```
SET INPSEL #10B1 2 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	None	INPSEL	#10B1	None	None	2	<LF>

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Input Choice	INPSEL	SET, GET, SUB	None	0 = mute 1..N = Chosen Input

4.5.11 Filter Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Bypass	FILBYP	SET, GET, SUB	None	0 = Filter active 1 = Filter inactive
Gain	FILGAIN	SET, GET, SUB, INC, DEC	None	-120..120 Stepwidth: 0.1 dB
Frequency	FILFREQ	SET, GET, SUB, INC, DEC	None	200..220000 Stepwidth: 0.1 dB

4.5.12 Crossover Block

Notice: Index1 is used to address the specific Crossover filter section to be configured. e.g:

In a 2-way Crossover:

- Index1 = 1: addresses the low-pass filter section
- Index1 = 2: addresses the high-pass filter section

In a 3-way Crossover:

- Index1 = 1: addresses the low-pass filter section
- Index1 = 2: addresses the mid filter section
- Index1 = 3: addresses the high-pass filter section

In a 4-way Crossover:

- Index1 = 1: addresses the low-pass filter section
- Index1 = 2: addresses the lower mid filter section
- Index1 = 3: addresses the higher mid filter section
- Index1 = 4: addresses the high-pass filter section

The application of the filter frequency configuration depends on the filter section type:

- For the low-pass filter section only the low-pass filter applies, the high-pass frequency setting (CROSSFREQHP) will be ignored.
- For the mid filter sections both, the low-pass (LP) and high-pass (HP), filter frequencies apply
- For the high-pass filter section only the high-pass filter applies, the low-pass frequency setting (CROSSFREQLP) will be ignored

Parameter of Functional Block	**Attribute String**	**Applicable Commands**	**Applicable Index**	**Applicable Value Range**
Mute Out	CROSSMUTE	SET, GET, SUB	Index1	0 = Output unmuted 1 = Output muted
Gain Out	CROSSGAIN	SET, GET, SUB, INC, DEC	Index1	-120..120 Stepwidth: 0.1 dB
Frequency LP	CROSSFREQLP	SET, GET, SUB, INC, DEC	Index1	200..220000 Stepwidth: 0.1 dB
Frequency HP	CROSSFREQHP	SET, GET, SUB, INC, DEC	Index1	200..220000 Stepwidth: 0.1 dB

Example: Command string to set the output gain for the mid filter section of a 3-way crossover to -20 dB:

```
SET CROSSGAIN #22B1 1 -200 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	None	CROSSGAIN	#22B1	1	None	-200	<LF>

4.5.13 Equalizer Block

In Equalizer commands, Index1 is used to address band parameters. The value of Index1 states the number of the EQ band for which the parameter is to be set or retrieved.

Example: String to set the gain for EQ band 4 to 10 dB.

Notice: The Instance ID in this example refers to the third EQ in a stack.

```
SET EQBGAIN #4B3 4 100 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SET	None	EQBGAIN	#4B3	4	None	100	<LF>

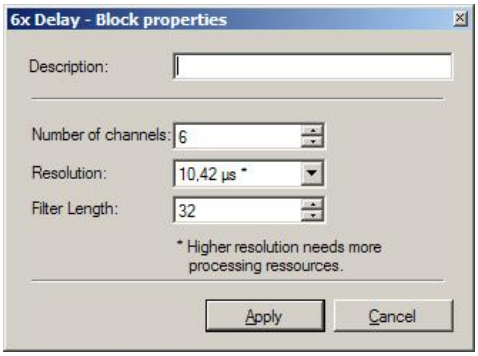
Parameter of Functional Block	**Attribute String**	**Applicable Commands**	**Applicable Index**	**Applicable Value Range**
Bypass All	EQBYPALL	SET, GET, SUB	None	0 = EQ active 1 = EQ inactive
Band Bypass	EQBBYP	SET, GET, SUB	Index1	0 = Band active 1 = Band inactive
Band Gain	EQBGAIN	SET, GET, SUB, INC, DEC	Index1	-120..120 Stepwidth: 0.1 dB
Band Frequency	EQBFREQ	SET, GET, SUB, INC, DEC	Index1	200..220000 Stepwidth: 0.1 dB

4.5.14 Delay Block

Parameter of Functional Block	**Attribute String**	**Applicable Commands**	**Applicable Index**	**Applicable Value Range**
Delay time	DELTIME	SET, GET, SUB	None	0..48000 [Samples] depending on configured resolution (see table above)
Bypass	DELBYP	SET, GET, SUB	None	0 = delay active 1 = delay inactive

NOTICE: Delay time is configured in terms of audio samples to achieve the maximum possible accuracy. To get the number of samples from a desired timespan in milliseconds the following formula applies: `Samples = Milliseconds ÷ Resolution.`

The accepted values for the Delay time attribute depend on the Resolution and Filter Length (FL) which may be configured in the properties dialog of the Delay block. I.e. for higher resolutions than the default 48 kHz delay values below FL (or 2×FL) are restricted to multiples of 2 (or 4) as displayed in the following table.



Resolution	**Value Range**
48 kHz ≈ 20,83 µs	{0..48000} (Note: This ist he default setting)
96 kHz ≈ 10,42 µs	{0, 2, .. FL-2}, {FL, FL+1,..,48000}
192 kHz ≈ 5,21 µs	{0, 4, .. 2×FL-4}, {2×FL, 2×FL+1,..,48000}

4.5.15 Generator Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Type	GENTYPE	SET, GET, SUB	None	0 = Sine 1 = Pink Noise 2 = White Noise
Level	GENLVL	SET, GET, SUB, INC, DEC	None	-1000..+100 Stepwidth: 0.1 dB
Mute	GENMUTE	SET, GET, SUB	None	0 = unmuted 1 = muted
Frequency	GENFREQ	SET, GET, SUB, INC, DEC	None	200..220000 Stepwidth: 0.1 Hz

4.5.16 FlipFlop Block

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
State	FFSTATE	SET, GET, SUB	None	0 = Off 1 = On

4.5.17 DMX Block

The DMX Control block may be used to control external DMX devices.

Index1 is used to address a specific DMX channel in the range of 1..512

Parameter of Functional Block	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Enable	DMXEN	SET, GET, SUB	None	0 = Controller disabled 1 = Controller enabled
Value	DMXVAL	SET, GET, SUB, INC, DEC	Index1	0..255

4.5.18 Presets

For Preset commands the Preset-ID is used as instance ID (see section 4.6).

Affected Parameters	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
All block parameters selected within Preset	PRESET	RECALL, STORE	None	None

4.5.19 System

System commands are applied system wide and don’t require a specific Instance ID (see section 4.6).

Parameter	Attribute String	Applicable Commands	Applicable Index	Applicable Value Range
Device Number	DEVNUM	SET, GET, SUB	None	0..255

4.6 H-Text: Instance ID

InstanceID identifies the item (e.g. Block or Preset) to be addressed.

HARVEY processing blocks are identified by the block and channel number which in combination make up the Instance ID. Both numbers are concatenated with the delimiter ‘B’ and prefixed with a ‘#’. That ID is displayed in HARVEY Composer in the IDs tab or at the bottom right corner of a Channel of an unfolded Block on the worksheet.

Example: “#1B4” identifies Block 1 channel 4.

Notice: In a Gang only one channel has to be addressed to set a value in all members of the gang.

Notice: Preset IDs may be in the Range of 0 .. max Preset ID. They may be retrieved from the ID column within the Presets Tab in HARVEY Composer.

4.7 H-Text: Index

Some attributes may hold different values when referring to different inputs, outputs or parameter sets of a block. Those values are addressed using one or both of the Index fields in a command string.

Examples:

- An Equalizer uses Index1 to address which band a setting corresponds to.
- A Matrix Mixer uses Index1 to address inputs and Index2 to address outputs and both in combination to address nodes within the matrix.

The usage of an Index will be indicated in the attribute definitions in section 4.5.

Notice: Index values start at 1. Therefore, in an equalizer block Index1 = 1 addresses the first band, Index1 = 2 the second band and so on.

4.8 H-Text: Value

The format of a value in a SET command depends on the Attribute to be set.

Numeric values are always represented as Integers (signed or unsigned). Value ranges may contain a step width which defines how the value is being interpreted.

Example:

The following command will set the Level Gain parameter of Block 3B12 to -44.3 dB.

```
SET 1 LVLGAIN #3B12 -443 <LF>
```

Notice: Level Gain has a step width of 0.1 dB.

4.9 H-Text: Responses

HARVEY responds with for different types of response messages.

Success: Upon successful execution of a SET, INC, DEC or RECALL command, HARVEY will respond with:

```
OK<CR><LF>
```

Value: Upon successful execution of a GET command, HARVEY will respond with the requested Attribute value, using the representation as specified in section 4.5, followed by carriage return and line feed.

Busy: If HARVEY is receiving messages faster than it can handle, it will drop new messages until enough resources for the handling of new messages are available and respond with:

```
BUSY<CR><LF>
```

Error: If execution of GET, SET, INC, DEC or RECALL command failed, HARVEY will respond with:

```
ERR<CR><LF>
```

Notice: HARVEY does not respond on received messages which it cannot interpret as H-Text commands.

4.10 H-Text: Subscriptions / Notifications

As of firmware version 1.9 the HARVEY-Device can send asynchronous notifications of parameter value changes. Notifications can be requested using the SUB command.

Upon value change a NFY message will be sent by the device. NFY messages follow syntax similar to the SET messages.

SUB commands follow a syntax similar to the GET command.

Notifications are considered only useful on peer-to-peer connections (e.g. RS232, TCP, UDP), thus the Device Number is omitted in SUB and NFY messages.

Example: Subscribe to Level Gain value of Block #3 channel 1:

```
SUB LVLGAIN #2B5 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
SUB	*Blank*	LVLGAIN	#2B5	*blank*	*blank*	*blank*	<LF>

The subscription will be confirmed with an Ok/Busy/Error response.

NFY messages will be sent initially for each value on subscribe and after that upon value change:

```
NFY LVLGAIN #2B5 -36 <LF>
NFY LVLGAIN #2B5 -74 <LF>
NFY LVLGAIN #2B5 -444 <LF>
```

The UNSUB command is used to cancel a previously executed subscription.

Example: Unsubscribe Level Gain value of Block #3 channel 1:

```
UNSUB LVLGAIN #2B5 <LF>
```

Command	Dev Nr	Attribute	Inst ID	Index1	Index2	Value	End
UNSUB	*Blank*	LVLGAIN	#2B5	*blank*	*blank*	*blank*	<LF>

There is no need to execute UNSUB before closing a network connection. Connection loss will automatically trigger unsubscribe of all subscriptions made by that connection. Thus, a client has to repeat all subscriptions after reconnecting.

5 Addendum

5.1 Electrical Interfaces

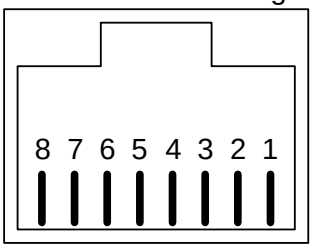
5.1.1 RS-485 (HARVEY mx.16)

HARVEY mx.16 has two RS-485 interfaces on its rear side represented by two 8-pin RJ45 connectors. Both interfaces are identical and allow a daisy-chained cabling of more than one device attached to HARVEY.

For long cable distances and/or high bit rates it is advised to use a terminating resistor of 120 Ohm at each end of a RS-485 line allowing a stable data link.

For RS485 system wiring please consider the advices in section 5.2.

The pinning of both RJ45 connectors is identical as following:



Type	**8-pin RJ45**
DATA+	Pin 3 + Pin 6
DATA-	Pin 4 + Pin 5
ISO GND	Pin 7 + Pin 8
SHIELD	CASE / PE

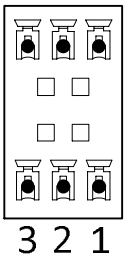
All other pins **must not** be used.

5.1.2 RS-485 (HARVEY Pro)

HARVEY Pro units have two RS-485 interfaces on its rear side represented by two 3-pin Phoenix connectors. Both interfaces are identical and allow a daisy-chained cabling of more than one device attached to HARVEY.

For long cable distances and/or high bit rates it is advised to use a terminating resistor of 120 Ohm at each end of a RS-485 line allowing a stable data link.

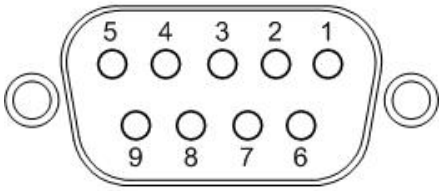
For RS485 system wiring please consider the advices in section 5.2.



Type	**3-pin PHOENIX (3.81 mm)**
DATA+	Pin 1
DATA-	Pin 3
ISO GND	Pin 2

5.1.3 RS-232

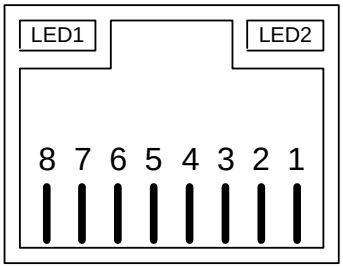
HARVEY has one RS-232 DCE interface on its rear side represented by a standard DB-9 female connector.



Type	Direction	DB-9 female
TXD	Out	Pin 2
RXD	In	Pin 3
CTS	In	Pin 7
RTS	Out	Pin 8
GND		Pin 5
SHIELD		CASE

5.1.4 10/100BaseT Ethernet

HARVEY has one 10/100BaseT Ethernet interface on its rear side represented by a standard RJ-45 connector including two LEDs displaying Ethernet link and activity.



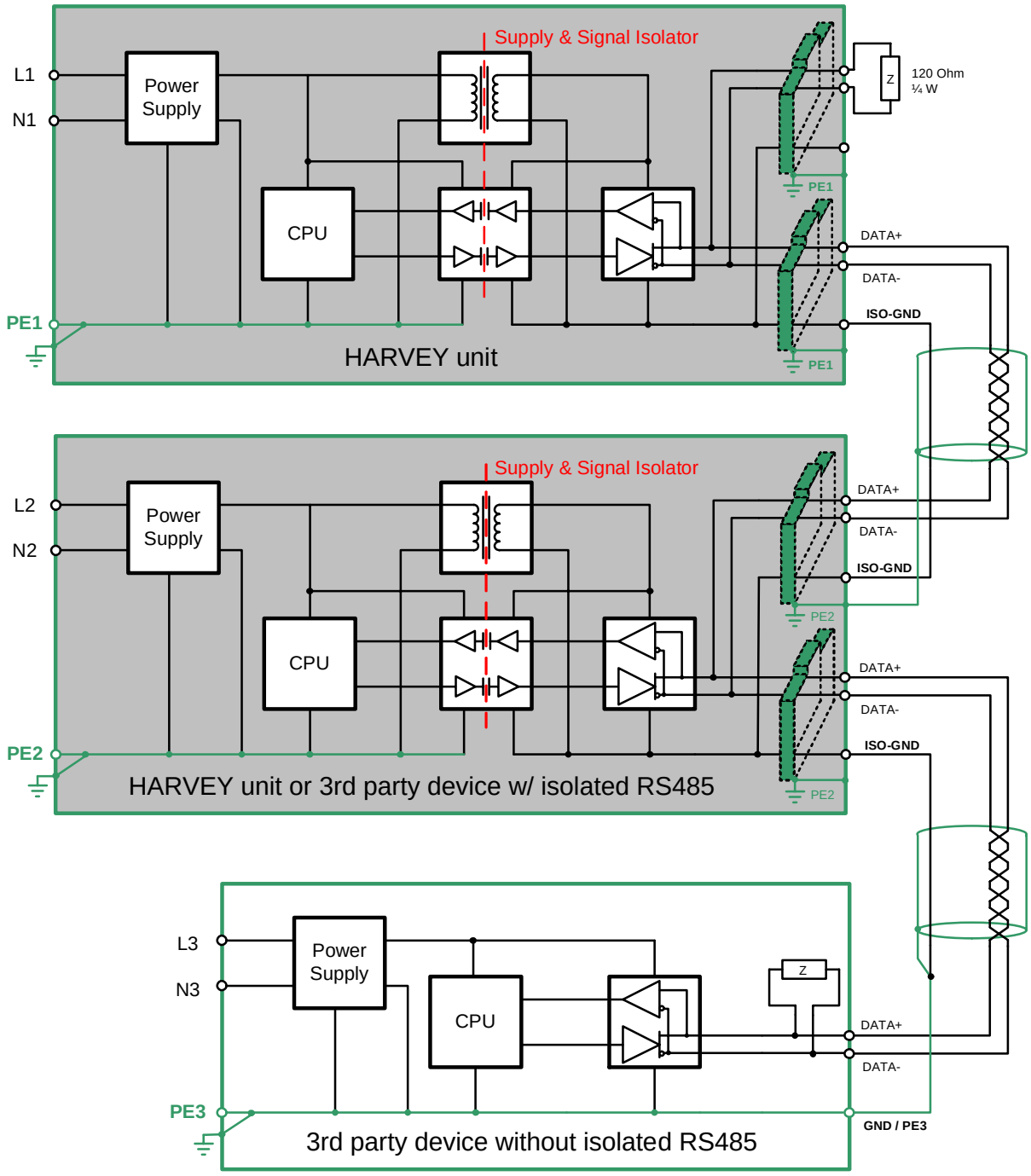
Type	Direction	8-pin RJ45
TXD+	Out	Pin 1
TXD-	Out	Pin 2
RXD+	In	Pin 3
RXD-	In	Pin 6
SHIELD		CASE

LED1: Activity / LED2: Link

5.2 RS-485 Wiring

HARVEY offers two 2-wire RS-485 interfaces carried out as two RJ45 connectors on its rear side. Internally both interfaces are connected to one data transceiver. Thus, a daisy-chained system integration of HARVEY RS-485 interface is made possible.

Following diagram shows (a) the internal structure of the RS-485 interface and (b) presents a best practice system wiring in a three device setup:



Internally the RS-485 circuit is carried out isolated with regard to signals and power supply. This structure avoids ground loops which may occur in systems with long distances between the protective earth (PE) -grounding points and which may lead to communication faults.
Each RS-485 interface of HARVEY offers a ground signal which is isolated from PE-ground. Only the RJ45-shield is connected to the PE-grounded device housing.

Thus, following best practice RS-485 wiring applies:

1. Only connect one side of a shielded twisted-pair cable to the RJ45-shield of the RS-485-interface to prevent ground loops but to keep the shield effect against EMI problems.

2. In addition to the connection of the data signals, connect the isolated ground signal of HARVEY RS-485 interconnections to each other.

3. If possible, use only one non-isolated RS-485 device and all other devices with an isolated RS-485 and connect the isolated ground signal to the PE-ground system only at one point in the system.

4. Use the passive terminator, which is delivered with HARVEY, and plug it into one RS-485 interface of HARVEY at the end of a RS-485-line which as a result allows bitrates of 500 kbps on cable distances >400 m.